Fast Algorithm for Logistic Bandit

Jun-Kun Wang
National Taiwan University
wangjim123@gmail.com

Chi-Jen Lu
Academia Sinica
cjlu@iis.sinica.edu

Shou-De Lin
National Taiwan University
sdlin@csie.ntu.edu.tw

Abstract

We study a logistic bandit problem and propose an algorithm that enjoys fast update. In our problem, each round the learner first chooses an arm from a decision set, in which each arm is associated with a feature vector. Then, she receives a reward, which is binary and is generated by a logistic function. Our algorithm for the problem can be seen as a marriage between stochastic gradient descent in optimization with confidence ball strategy in stochastic linear bandit literature. If the decision space is finite k arms with d dimensional feature vectors, our algorithm enjoys $\mathcal{O}(kd)$ computational complexity in each round, which is better than $\mathcal{O}(kd^2)$ of related works. We give some theoretical analysis of the proposed algorithm for the regret upper bound. Furthermore, we consider a distributed bandit setting such that there are some learners conducting the online learning. We extend our algorithm to the distributed setting. By communication, the learners can achieve speedup in learning, which is measured by regret. We also conduct experiments on two recommendation datasets (MovieLens and Yahoo! Front Page) to show that our algorithm not only updates faster but also achieves highly competitive click-through rate with the baseline.

Introduction

Online learning algorithms have drawn growing interests because of their theoretical guarantees and their applications in sequential learning and decision making. Among these algorithms, stochastic linear bandit (e.g. (Auer 2002; Dani, Hayes, and Kakade 2008; Li et al. 2010; Abbasi-yadkori et al. 2011)) has been adopted in web recommendation systems and shown to have some success. In the setting, in each round, the learner first observes some contexts (feature vectors) associated with some items. The learner then selects an item according to her decision rule. After that, the reward of choosing the item is revealed, while the rewards for choosing the others remain hidden. Finally, the learner updates her model based on her previous decisions and observations and then continues to the next round. The setting is so general that the related algorithms for the bandit problem have been applied to web advertising and recommendation (Li et al. 2010; Li et al. 2011; Abbasi-yadkori et al. 2011). However, previous works for stochastic linear bandit has a computational issue. Their algorithms update slowly when operating on a high dimensional context space. This is because the algorithms have to maintain the inverse of a matrix and the online update of the inverse matrix requires $\mathcal{O}(d^2)$ computations by Sherman–Morrison formula. Moreover, when choosing an arm, the algorithms involve some multiplications of the maintained matrix and the feature vector of each arm, which is $\mathcal{O}(kd^2)$ in general. This prevents from using rich feature representation of items. Thus, the performance for choosing good items is sacrificed for reducing response/update time and saving the computations.

In this work, we address this issue and propose an efficient algorithm; the complexity of update in the proposed algorithm scales linearly with the dimension of a context space. We achieve this by considering an assumption of reward which is different from the one in the original stochastic linear bandit problem. Previous works (e.g. (Auer 2002; Dani, Hayes, and Kakade 2008; Li et al. 2010; Abbasi-yadkori et al. 2011)) assume the expected reward of making a decision is the inner product of a unknown vector and the feature vector that corresponds to the decision. Instead, we study a logistic banidt problem, which was recently proposed by (Zhang et al. 2016). The reward is generated by a logistic model. It is binary and the probabilities of the outcome is determined by the inner product of a unknown vector and the context vector associated with the decision through the logistic function.

We also extend our algorithm to a distributed bandit setting, which is another major contribution in this paper. The setting is that there are m learners doing online learning. The learners can communicate with a master to exchange their information. By communication, the learners can improve their learning and predicting performance. Our proposed algorithm allows the learners to achieve $\tilde{\mathcal{O}}(\sqrt{T/m})$ regret under certain condition, compared to $\tilde{\mathcal{O}}(\sqrt{T})$ when learning alone, where T is the number of rounds. We believe that the proposed distributed algorithm is useful for a recommendation system, as such a system usually needs to provide service to many users online at the same time.

Our results. We propose an algorithm that for logistic bandit. The computational complexity is $\mathcal{O}(kd)$, which is a factor of $\mathcal{O}(d)$ improvement over the related works. We prove that the algorithm has $\mathcal{O}(\sqrt{T \log T})$ regret. Most im-

portantly, the dimension of the context space does not appear explicitly in the regret bound. We also study the distributed bandit setting and propose a distributed algorithm that allows the learner to achieve speedup in learning. We conduct experiments and compare our algorithm with a popular stochastic linear bandit algorithm, which shows that our algorithm not only runs faster than the baseline but also achieves highly competitive click-through rate (CTR) with the baseline.

Preliminaries

As discussed in the introduction, we are interested in a specific stochastic linear bandit problem, which has recently been considered in (Zhang et al. 2016). In each round, the learner first makes a decision $\mathbf{x}_t \in \mathbb{R}^d$ from a decision set $\mathcal{D} \in \mathbb{R}^d$. Then, she receives a reward $r_t \in \mathbb{R}$. The reward is assumed to be binary $r_t \in \{0, 1\}$ and is generated from the logistic model,

$$Pr[r_t = 1|\mathbf{x}_t] = \frac{1}{1 + \exp(-\mathbf{x}_t^\top \mathbf{w}^*)} = \frac{\exp(\mathbf{x}_t^\top \mathbf{w}^*)}{1 + \exp(\mathbf{x}_t^\top \mathbf{w}^*)}, \tag{1}$$

and $Pr[r_t = 0|\mathbf{x}_t] = 1 - Pr[r_t = 1|\mathbf{x}_t]$, where $\mathbf{w}^* \in \mathbb{R}^d$ is a unknown vector. If $\mathbf{x}_t^\top \mathbf{w}^*$ is large, then the probability that observing the reward 1 is high, as the function is monotone increasing with respect to the parameter $\mathbf{x}_t^\top \mathbf{w}^*$. The assumption of rewards can model the click ($r_t = 1$) or no-click ($r_t = 0$) of an advertisement ($\mathbf{x}_t$) in web advertising. For web advertising, one would like to design an algorithm to maximize the number of user clicks over time. Due to the assumption of rewards, the conditional expected number of clicks achieved by an algorithm would be $\Sigma_{t=1}^T \exp(\mathbf{x}_t^\top \mathbf{w}^*)/(1 + \exp(\mathbf{x}_t^\top \mathbf{w}^*))$. As our problem belongs to online learning, a common way to measure the performance of the learner is to compare the expected clicks she gets with the one by a clairvoyant who knows $\mathbf{w}^*$ in hindsight. The difference, which is called the *pseudo regret* of the learner, is

$$T \max_{x \in \mathcal{D}} \frac{\exp(\mathbf{x}^\top \mathbf{w}^*)}{1 + \exp(\mathbf{x}^\top \mathbf{w}^*)} - \sum_{t=1}^{T} \frac{\exp(\mathbf{x}_t^\top \mathbf{w}^*)}{1 + \exp(\mathbf{x}_t^\top \mathbf{w}^*)}. \tag{2}$$

However, we do not analyze the regret bound of an algorithm based on the definition above due to some technical difficulties. Instead, we provide an upper bound of the following measure,

$$T \max_{x \in \mathcal{D}} \mathbf{x}^\top \mathbf{w}^* - \Sigma_{t=1}^T \mathbf{x}_t^\top \mathbf{w}^*, \tag{3}$$

while we still use the assumption of the rewards when deriving the upper bound. One can show that (2) and (3) are at the same order. Denote the value of (2) as (A) and the value of (3) as (B). (Zhang et al. 2016) has shown that $\frac{1}{2(1+\exp(\theta))}$ (B) $\leq$ (A) $\leq \frac{1}{4}$ (B) , assuming $\|(\mathbf{w}^*)^\top \mathbf{x}\|_2 \leq \theta$ for any $\mathbf{x} \in \mathcal{D}$. Consequently, the derived upper bound of (3) is within a constant multiple of (2).

Algorithm

Let us begin by giving another assumption and notation. Without loss of generality, we assume for every $\mathbf{x} \in \mathcal{D}$ in the decision space, its L2 norm satisfies $\|\mathbf{x}\|_2 \leq u$. Moreover, $\mathbf{x}^\top \mathbf{w}^* \neq 0$ for all $\mathbf{x} \in \mathcal{D}$. In the following, for brevity, the probability of the binary reward $r_t = 0$ is mapped to $r_t = -1$ so that the probability (1) can be written into a compact form, i.e. $Pr[r_t = \{\pm 1\}|\mathbf{x}_t] = \frac{1}{1+\exp(-r_t \mathbf{x}_t^\top \mathbf{w}^*)} = \frac{\exp(r_t \mathbf{x}_t^\top \mathbf{w}^*)}{1+\exp(r_t \mathbf{x}_t^\top \mathbf{w}^*)}$. Let us denote $f_t(\mathbf{w}) = \log(1 + \exp(-r_t \mathbf{x}_t^\top \mathbf{w}))$, which is the logistic loss function. It follows that maximizing the probability function $\frac{1}{1+\exp(-r_t \mathbf{x}_t^\top \mathbf{w})}$ over $\mathbf{w}$ is equivalent to minimizing the logistic loss $f_t(\mathbf{w})$. Next, for each function $f_t(\cdot)$, let $\bar{f}_t(\mathbf{w}) = \mathbb{E}_{r_t}[\log(1 + \exp(-r_t \mathbf{x}_t^\top \mathbf{w})]$ be its conditional expectation over the reward r_t when choosing $\mathbf{x}_t$, where r_t is generated by the logistic model.

We also assume the unknown vector $\mathbf{w}^* \in \mathbb{R}^d$ is also a minimizer of

$$\min_{\mathbf{w}} \bar{f}_t(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2, \forall t, \tag{4}$$

for some suitable λ. To justify this assumption, we borrow the following lemma.

Lemma 1. *((Zhang et al. 2016))* $\bar{f}_t(\mathbf{w}) - \bar{f}_t(\mathbf{w}^*) = D_{KL}(p_{\mathbf{w}^*,\mathbf{x}_t}||p_{\mathbf{w},\mathbf{x}_t}) \geq 0$, *where* $D_{KL}(\cdot||\cdot)$ *means the KL-divergence,* $p_{\mathbf{w}^*,\mathbf{x}_t} \equiv \frac{1}{1+\exp(-\mathbf{x}_t^\top \mathbf{w}^*)}$ *is the bernoulli distribution induced by* $\mathbf{w}^*$, *and* $p_{\mathbf{w},\mathbf{x}_t} \equiv \frac{1}{1+\exp(-\mathbf{x}_t^\top \mathbf{w})}$ *is the one induced by* $\mathbf{w}$.

The lemma means $\mathbf{w}^*$ is the minimizer of $\min_{\mathbf{w}} \bar{f}_t(\mathbf{w}), \forall t$. Yet, to control the model complexity, we also require $\mathbf{w}^*$ satisfies the optimization problem (4). This holds when $\bar{f}_t(\mathbf{0})$ is larger than $\bar{f}_t(\mathbf{w}^*)$ by $\frac{\lambda}{2}\|\mathbf{w}^*\|_2^2$. Thus, we further assume that $\mathbf{w}^*$ is in a L2 norm ball whose radius is R, $\|\mathbf{w}^*\|_2 \leq R$. We think the condition is not strict at all, as long as $\mathbf{x}^\top \mathbf{w}^* \neq 0$ for all $\mathbf{x} \in \mathcal{D}$, since the model $\mathbf{w} = \mathbf{0}$ simply implies that each items gets a click (reward 1) with the same probability 0.5 and this implication is unlikely to be true in practice.

Our algorithm is shown in the following block, where Π_R is the projection into the ball, and $\nabla f_t(\mathbf{w}_t)$ is the gradient of the function $f_t(\mathbf{w}) = \log(1 + \exp(-r_t \mathbf{x}_t^\top \mathbf{w}))$ at point $\mathbf{w}_t$. The algorithm requires two parameters, which are the radius of a confidence ball γ_t and the learning rate η_t. Both parameters are defined in the following section.

Algorithm 1 Our algorithm

Require γ_t and η_t.
Initialization: Let $\mathbf{w_1} = \mathbf{0} \in \mathbb{R}^d$.
1: $(\mathbf{x}_t, \hat{\mathbf{w}}_t) = \arg\max_{\mathbf{x} \in \mathcal{D}, \mathbf{w} \in \mathcal{C}_t} \mathbf{x}^\top \mathbf{w}$,
where $C_t = \{\mathbf{w} : \|\mathbf{w} - \mathbf{w}_t\|_2 \leq \sqrt{\gamma_t}\}$.
2: Select $\mathbf{x}_t$ and observe a reward $r_t = 1$ or $r_t = -1$.
3: Update $\mathbf{w}_{t+1} = \Pi_R(\mathbf{w}_t - \eta_t(\nabla f_t(\mathbf{w}_t) + \lambda \mathbf{w}_t))$.

Clearly, the update (line 3 in Algorithm 1) is $\mathcal{O}(d)$. For the optimization problem in line 1, if the decision set $\mathcal{D}$ is finite: $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{|\mathcal{D}|}\}$, then $\mathbf{x}_t = \arg\max_{x_k} \mathbf{w}_t^\top \mathbf{x}_k + \sqrt{\gamma_t}\|\mathbf{x}_k\|_2$. That is, the total computational complexity is $\mathcal{O}(kd)$, which is better than $\mathcal{O}(kd^2)$ of many related works (e.g. (Abbasi-yadkori et al. 2011; Zhang et al. 2016)).

Theoretical analysis

In this section, we analyze the regret of Algorithm 1. As mentioned in the preliminaries section, our goal is to provide the upper bound of (3) for our algorithms.

The following theorem shows that with high probability, the squared distance between our algorithm's $\mathbf{w}_t$ and the unknown $\mathbf{w}^*$ in each round t is scaled with $\mathcal{O}(1/t)$.

Theorem 1. *Let $\delta \in (0, 1/e)$ and assume $T \geq 4$ and $\lambda \leq 1$. If $\eta_t = \frac{2}{\lambda t}$, then with probability at least $1-\delta$, we have $\|\mathbf{w}_t - \mathbf{w}^*\|_2^2 \leq \frac{(2304u^2+96uR)\log(T\log(T)/\delta)+24(u^2+\lambda^2R^2)}{\lambda^2 t}$, for any $t \leq T$, where u is the upper bound of the L2 norm of any $\mathbf{x} \in \mathcal{D}$ respectively.*

Proof. Denote a function $\psi_t(\cdot)$ as $f_t(\cdot) + \frac{\lambda}{2}\|\cdot\|_2^2$. The update $\mathbf{w}_{t+1} = \Pi_R(\mathbf{w}_t - \eta_t(\nabla f_t(\mathbf{w}_t) + \lambda \mathbf{w}_t))$ is equivalent to $\mathbf{w}_{t+1} = \Pi_R(\mathbf{w}_t - \eta_t \nabla \psi_t(\mathbf{w}_t))$. Now we bound the distance.

$\|\mathbf{w}_{t+1} - \mathbf{w}^*\|_2^2 = \|\Pi_R(\mathbf{w}_t - \eta_t \nabla \psi_t(\mathbf{w}_t)) - \mathbf{w}^*\|_2^2 \overset{(1)}{\leq} \|\mathbf{w}_t - \eta_t \nabla \psi_t(\mathbf{w}_t) - \mathbf{w}^*\|_2^2 = \|\mathbf{w}_t - \mathbf{w}^*\|_2^2 - 2\eta_t \langle \nabla \psi_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle + \eta_t^2 \|\nabla \psi_t(\mathbf{w}_t)\|_2^2 = \|\mathbf{w}_t - \mathbf{w}^*\|_2^2 - 2\eta_t \langle \nabla \bar{\psi}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle + \eta_t^2 \|\nabla \psi_t(\mathbf{w}_t)\|_2^2 + 2\eta_t \langle \nabla \bar{\psi}_t(\mathbf{w}_t) - \nabla \psi_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle \overset{(2)}{\leq} \|\mathbf{w}_t - \mathbf{w}^*\|_2^2 - 2\eta_t(\bar{\psi}_t(\mathbf{w}_t) - \bar{\psi}_t(\mathbf{w}^*) + \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{w}^*\|^2) + \eta_t^2 \|\nabla \psi_t(\mathbf{w}_t)\|_2^2 + 2\eta_t \langle \nabla \bar{\psi}_t(\mathbf{w}_t) - \nabla \psi_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle \overset{(3)}{\leq} (1 - \eta_t \lambda)\|\mathbf{w}_t - \mathbf{w}^*\|_2^2 + 2\eta_t \langle \nabla \bar{\psi}_t(\mathbf{w}_t) - \nabla \psi_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle + \eta_t^2(2u^2 + 2\lambda^2 R^2)$.

Above, (1) is by the known property of projection. (2) is due to the fact that any non-strongly convex function added by $\frac{\lambda}{2}\|\cdot\|_2^2$ becomes a λ strongly convex function. Thus, $\bar{\psi}_t(\mathbf{w}^*) \geq \bar{\psi}_t(\mathbf{w}_t) + \nabla \bar{\psi}_t(\mathbf{w}_t)^\top(\mathbf{w}^* - \mathbf{w}_t) + \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{w}^*\|_2^2$. Rearraging it leads to $-\langle \nabla \bar{\psi}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle \leq -(\bar{\psi}_t(\mathbf{w}_t) - \bar{\psi}_t(\mathbf{w}^*) + \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{w}^*\|_2^2)$. (3) is because of the assumption that $\mathbf{w}^*$ is the minimizer of $\bar{\psi}_t(\cdot) \equiv \bar{f}_t(\cdot) + \frac{\lambda}{2}\|\cdot\|_2^2$ so that $\bar{\psi}_t(\mathbf{w}_t) - \bar{\psi}_t(\mathbf{w}^*) \geq 0$ and the fact that $\|\nabla \psi_t(\mathbf{w}_t)\|_2^2 = \|\nabla f_t(\mathbf{w}_t) + \lambda \mathbf{w}_t\|_2^2 \leq 2(\|\nabla f_t(\mathbf{w}_t)\|_2^2 + \lambda^2 \|\mathbf{w}_t\|_2^2) \leq 2(u^2 + \lambda^2 R^2)$, as $\|\nabla f_t(\mathbf{w}_t)\|^2 = (\frac{\exp(-y_t \mathbf{x}_t^\top w)}{1+\exp(-y_t \mathbf{x}_t^\top w)})^2 \mathbf{x}_t^\top \mathbf{x}_t \leq \|\mathbf{x}_t\|_2^2 \leq u^2$.

Now we have derived that $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|_2^2 \leq (1 - \eta_t\lambda)\|\mathbf{w}_t - \mathbf{w}^*\|_2^2 + 2\eta_t \langle \nabla \bar{f}_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle + \eta_t^2(2u^2 + 2\lambda^2 R^2)$. Unwinding the derived inequality till $t = 2$, we get $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|_2^2 \leq \frac{4}{\lambda}\Sigma_{i=2}^t \frac{1}{i}(\Pi_{j=i+1}^t (1 - \frac{2}{j}))\langle \nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i), \mathbf{w}_i - \mathbf{w}^* \rangle + \frac{4}{\lambda^2}\Sigma_{i=2}^t \frac{1}{i^2}(\Pi_{j=i+1}^t (1 - \frac{2}{j}))(2u^2 + 2\lambda^2 R^2)$. Then, by using the facts that $\Pi_{j=i+1}^t (1 - \frac{2}{j}) = \Pi_{j=i+1}^t (\frac{j-2}{j}) = \frac{(i-1)i}{(t-1)t}$ and $\Sigma_{i=2}^t \frac{1}{i^2}\Pi_{j=i+1}^t (1 - \frac{2}{j}) = \Sigma_{i=2}^t \frac{(i-1)}{i(t-1)t} \leq \frac{1}{t}$, we have

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \frac{4}{\lambda^2 t}(2u^2 + 2\lambda^2 R^2) + \frac{4}{\lambda t(t-1)}\Sigma_{i=2}^t (i-1)\langle \nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i), \mathbf{w}_i - \mathbf{w}^* \rangle. \tag{5}$$

We can continue to provide the bound of the distance. The process is similar as the proof for Proposition 1 in (Rakhlin, Shamir, and Sridharan 2012). Observe that $Z_i = (i-1)\langle \nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i), \mathbf{w}_i - \mathbf{w}^* \rangle$ for each i is a martingale difference sequence, The conditional expectation given the previous rounds is 0. Furthermore, $|Z_i| \leq (t-1)\|\nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i)\|_2 \|\mathbf{w}_i - \mathbf{w}^*\|_2 \leq 2(t-1)u\|\mathbf{w}_i - \mathbf{w}^*\|_2 \leq 4(t-1)uR$. Thus, the expected value of Z_i is bounded. Let $\mathcal{F}_{i-1}$ be the randomness up to round $i-1$. The conditional variance $\mathrm{Var}[Z_i|\mathcal{F}_{i-1}]$ is bounded by $(i-1)^2\|\nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i)\|_2^2 \|\mathbf{w}_i - \mathbf{w}^*\|_2^2 \leq 4u^2(i-1)^2\|\mathbf{w}_i - \mathbf{w}^*\|_2^2$, using that fact that $\mathrm{Var}[\cdot] \leq \mathbb{E}[(\cdot)^2]$ and Cauchy-Schwarz inequality.

Then, we can follow (Rakhlin, Shamir, and Sridharan 2012) using the lemma below, which is a variant of Freeman's inequality.

Lemma 2. *(Lemma 3 in (Rakhlin, Shamir, and Sridharan 2012)) Let $Z_1, \ldots, Z_T$ be a martingale difference sequence with a uniform bound $|Z_i| \leq b$ for all i. Let $V_s = \Sigma_{t=1}^s Var_{t-1}(Z_t)$ be the sum of conditional variance of Z_t's. Further, let $\sigma_s = \sqrt{V_s}$. Then we have, for any $\delta \leq \frac{1}{e}$ and $T \geq 4$, $Pr(\Sigma_{t=1}^s Z_t \geq 2\max(2\sigma_s, b\sqrt{\log(1/\delta)})\sqrt{\log(1/\delta)}$ for some $s \leq T) \leq \log(T)\delta$*

By using the above analysis and Lemma 2, we have $\Sigma_{i=2}^t (i-1)\langle \nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i), \mathbf{w}_i - \mathbf{w}^* \rangle = \Sigma_{i=2}^t Z_i \leq 2\max(4u\sqrt{\Sigma_{i=2}^t (i-1)^2\|\mathbf{w}_i - \mathbf{w}^*\|_2^2}, 4uR(t-1)\sqrt{\log(\frac{T\log(T)}{\delta})})\sqrt{\log(\frac{T\log(T)}{\delta})}$ for all $t \leq T$ with probability $1-\delta$. Substituting it into (5) leads to $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \frac{32}{\lambda t(t-1)}\max(u\sqrt{\Sigma_{i=2}^t (i-1)^2\|\mathbf{w}_i - \mathbf{w}^*\|_2^2}, uR(t-1) \times \sqrt{\log(\frac{T\log(T)}{\delta})})\sqrt{\log(\frac{T\log(T)}{\delta})} + \frac{4}{\lambda^2 t}(2u^2 + 2\lambda^2 R^2) \leq \frac{32u\sqrt{\log(\frac{T\log(T)}{\delta})}}{\lambda t(t-1)}\sqrt{\Sigma_{i=2}^t (i-1)^2\|\mathbf{w}_i - \mathbf{w}^*\|_2^2} + \frac{1}{\lambda^2 t}(32uR\log(\frac{T\log(T)}{\delta}) + 8u^2 + 8\lambda^2 R^2)$, assuming $\lambda \leq 1$.

Now let $m = \frac{32u\sqrt{\log(\frac{T\log(T)}{\delta})}}{\lambda}$, $n = \frac{1}{\lambda^2}(32uR\log(\frac{T\log(T)}{\delta}) + 8u^2 + 8\lambda^2 R^2)$. We can rewrite it as $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \frac{m}{t(t-1)}\sqrt{\Sigma_{i=2}^t (i-1)^2\|\mathbf{w}_i - \mathbf{w}^*\|_2^2} + \frac{n}{t}$. Using mathematical induction by assuming $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \frac{a}{t+1}$ and finding a, we can derive the theorem. That is, we need to find an a so that

$$\frac{a}{t+1} \geq \frac{m}{t(t-1)}\sqrt{\Sigma_{i=2}^t (i-1)^2 \frac{a}{i}} + \frac{n}{t}.$$

It follows that a should satisfy $a \geq \frac{9m^2}{4} + 3n$. Substituting the definition of m and n into a and observing that the base case $t = 1$ also satisties the inequality, we get $\|\mathbf{w}_t - \mathbf{w}^*\|_2^2 \leq \frac{(2304u^2+96uR)\log(T\log(T)/\delta)+24(u^2+\lambda^2R^2)}{\lambda^2 t}$, □

Theorem 2. *Let $\delta \in (0, 1/e)$ and assume $T \geq 4$ and $\lambda \leq 1$. Set $\gamma_t \leq \frac{(2304u^2+96uR)\log(T\log(T)/\delta)+24(u^2+\lambda^2R^2)}{\lambda^2 t}$, Then, with probability at least $1-\delta$, our algorithm achieves $T\max_{x \in \mathcal{D}} \mathbf{x}^\top \mathbf{w}^* - \Sigma_{t=1}^T \mathbf{x}_t^\top \mathbf{w}^* \leq 4\sqrt{\frac{(2304u^4+96u^3R)\log(T\log(T)/\delta)+24(u^4+\lambda^2u^2R^2)}{\lambda^2}\log(T)T}$.*

Proof. Let $\mathbf{x}^*$ be the optimum of $\max_{x\in\mathcal{D}} \mathbf{x}^\top \mathbf{w}^*$. Then $T\max_{x\in\mathcal{D}} \mathbf{x}^\top \mathbf{w}^* - \Sigma_{t=1}^T \mathbf{x}_t^\top \mathbf{w}^* = \Sigma_{t=1}^T \mathbf{x}^{*\top} \mathbf{w}^* - \mathbf{x}_t^\top \mathbf{w}^* \overset{(1)}{\leq} \Sigma_{t=1}^T \mathbf{x}_t^\top \hat{\mathbf{w}}_t - \mathbf{x}_t^\top \mathbf{w}^* = \Sigma_{t=1}^T \mathbf{x}_t^\top (\hat{\mathbf{w}}_t - \mathbf{w}_t) + \mathbf{x}_t^\top (\mathbf{w}_t - \mathbf{w}^*) \leq \Sigma_{t=1}^T \|\mathbf{x}_t\|_2 (\|\hat{\mathbf{w}}_t - \mathbf{w}_t\|_2 + \|\mathbf{w}_t - \mathbf{w}^*\|_2) \overset{(2)}{\leq} \Sigma_{t=1}^T \|\mathbf{x}_t\|_2 (\sqrt{\gamma_t} + \sqrt{\gamma_t})$, where (1) is due to the optimization problem in line 1 in Algorithm 1. (2) is because $\|\hat{\mathbf{w}}_t - \mathbf{w}_t\|_2 \leq \sqrt{\gamma_t}$ holds due to the constraint in line 1 of the algorithm. Moreover, the term $\|\mathbf{w}_t - \mathbf{w}^*\|_2$ is bounded by $\sqrt{\gamma_t}$ for all t with probability $1-\delta$, according to Theorem 1.

Thus, we have $T\max_{x\in\mathcal{D}} \mathbf{x}^\top \mathbf{w}^* - \Sigma_{t=1}^T \mathbf{x}_t^\top \mathbf{w}^* \leq 2\Sigma_{t=1}^T \sqrt{\gamma_t} \|\mathbf{x}_t\|_2 \leq 2\sqrt{\Sigma_{t=1}^T \gamma_t} \sqrt{\Sigma_{t=1}^T \|\mathbf{x}_t\|_2^2} \leq 2\sqrt{u^2 T} \sqrt{\Sigma_{t=1}^T \gamma_t} \leq 2\sqrt{u^2 T} \sqrt{\Sigma_{t=1}^T \frac{(2304u^2+96uR)\log(T\log(T)/\delta)+24(u^2+\lambda^2R^2)}{\lambda^2 t}} \leq 4\sqrt{\frac{(2304u^4+96u^3R)\log(T\log(T)/\delta)+24(u^4+\lambda^2u^2R^2)}{\lambda^2}\log(T)T}$, where the second inequality is by using Cauchy-Schwarz inequality. □

Theorem 2 means that Algorithm 1 can achieve $\mathcal{O}(\sqrt{T\log T})$ regret of (2). The upper bound does not depend explicitly on the dimension of the context space, d. The dimension is implicitly connected to the bound through the L2 norm assumption of the decision set, namely, $\|\mathbf{x}\|_2^2 \leq u^2, \forall \mathbf{x}$. This regret upper bound is $\mathcal{O}(d)$ improvement over the one by (Zhang et al. 2016). The computational complexity is also $\mathcal{O}(d)$ improvement over (Zhang et al. 2016).

We note that the above theoretical analysis assumes the decision space $\mathcal{D}$ is fixed in each round. Yet, the assumption is not necessary. If the decision space is changed over rounds ($\mathcal{D}_t$ instead of $\mathcal{D}$), the analysis can still proceed.

Distributed logistic bandit

In this section, we extend our algorithm to a distributed scenario. The motivation is that in a typical recommendation system, there are many users interacting with the system at the same time. Specifically, we consider the scenario that there are m learners; each provides recommendation to a user at a time. By communication, the learners can improve their learning and predicting performance.

We assume the distributed architecture is that their exists a master communicating with the m learners. The master maintains and updates a global parameter, while a learner uses a global parameter to provide recommendation and receives the feedback from a user. Our assumption for the communication protocol adopts the cyclic delayed update fashion (round-robin fashion), which has been considered in (Langford, Smola, and Zinkevich 2009; Agarwal and Duchi 2011) in the distributed optimization literature.

Our distributed algorithm for the learners and the master are shown on Algorithm 2 and Algorithm 3 respectively. The master maintains a global index t. At each t, the master communicates with a learner in the cyclic fashion and exchanges the information. It is the master that performs the update of the global parameter $\mathbf{w}$, while a learner interacts with a user, makes a decision, receives the feedback, and computes the gradient. The master performs the updates by using out-of-date information (last line in Algorithm 3) due to the communication protocol. Because of the cyclic delayed protocol, the gradient used for the update at step t is computed from the global parameter obtained at step $t-m$. For step 1 to step m, one can execute Algorithm 1 to initialize the master and the learners.

Algorithm 2 Distributed algorithm (learner)

1: Receive $\mathbf{v}$ and θ from the master
2: $(\mathbf{x}, \hat{\mathbf{w}}) = \arg\max_{\mathbf{x}\in\mathcal{D}, \mathbf{w}\in\mathcal{C}} \mathbf{x}^\top \mathbf{w}$,
where $C = \{\mathbf{w} : \|\mathbf{w} - \mathbf{v}\|_2 \leq \sqrt{\theta}\}$.
3: Select $\mathbf{x}$ and observe reward $r = 1$ or $r = -1$.
4: Compute $\nabla g(\mathbf{v}) = \nabla \log(1 + \exp(-r\mathbf{x}^\top \mathbf{v}))$.
5: Send $\nabla g(\mathbf{v}) + \lambda \mathbf{v}$ when the master calls it again.

Algorithm 3 Distributed algorithm (master)

For $t = m+1, \ldots, T$
1: Communicate with a learner (in the cyclic fashion).
2: Send $\mathbf{v} = \mathbf{w}_t$ and $\theta = \gamma_t$ to the learner
3: Receive $\nabla \psi_{t-m}(\mathbf{w}_{t-m}) = \nabla g(\mathbf{w}_{t-m}) + \lambda \mathbf{w}_{t-m}$ from the learner.
4: Update $\mathbf{w}_{t+1} = \Pi(\mathbf{w}_t - \eta_t \nabla \psi_{t-m}(\mathbf{w}_{t-m}))$

In the following, we analyze our distributed algorithm.

Theorem 3. *Let $\delta \in (0, 1/e)$ and assume $T \geq 4$ and $\lambda \leq 1$. If $\eta_t = \frac{2}{\lambda t}$, then with probability at least $1-\delta$, we have $\|\mathbf{w}_t - \mathbf{w}^*\|_2^2 \leq \frac{(2304u^2+96uR)\log(T\log(T)/\delta)+24(u^2+\lambda^2R^2)}{\lambda^2 t} + \frac{12Ru\sqrt{2u^2+2\lambda^2R^2}m(\log m+2)}{\lambda^2 t}$ for any $t \leq T$, where u is the upper bound of the L2 norm of any $\mathbf{x} \in \mathcal{D}$ respectively.*

Proof. $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|_2^2 = \|\Pi_R(\mathbf{w}_t - \eta_t \nabla \psi_{t-m}(\mathbf{w}_{t-m})) - \mathbf{w}^*\|_2^2 \leq \|\mathbf{w}_t - \eta_t \nabla \psi_{t-m}(\mathbf{w}_{t-m}) - \mathbf{w}^*\|_2^2 = \|\mathbf{w}_t - \mathbf{w}^*\|_2^2 - 2\eta_t \langle \nabla \psi_{t-m}(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^* \rangle + \eta_t^2 \|\nabla \psi_{t-m}(\mathbf{w}_{t-m})\|_2^2 = \|\mathbf{w}_t - \mathbf{w}^*\|_2^2 - 2\eta_t \langle \nabla \psi_{t-m}(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle + 2\eta_t \langle \nabla \psi_{t-m}(\mathbf{w}_t) - \nabla \psi_{t-m}(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^* \rangle + \eta_t^2 \|\nabla \psi_{t-m}(\mathbf{w}_{t-m})\|_2^2 = \|\mathbf{w}_t - \mathbf{w}^*\|_2^2 - 2\eta_t \langle \nabla \psi_{t-m}(\mathbf{w}_t) - \nabla \bar{\psi}_{t-m}(\mathbf{w}_t) + \nabla \bar{\psi}_{t-m}(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle + 2\eta_t \langle \nabla \psi_{t-m}(\mathbf{w}_t) - \nabla \psi_{t-m}(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^* \rangle + \eta_t^2 \|\nabla \psi_t(\mathbf{w}_{t-m})\|_2^2 \overset{(1)}{\leq} (1 - \eta_t\lambda)\|\mathbf{w}_t - \mathbf{w}^*\|_2^2 - 2\eta_t \langle \nabla \psi_{t-m}(\mathbf{w}_t) - \nabla \bar{\psi}_{t-m}(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle + 2\eta_t \langle \nabla \psi_{t-m}(\mathbf{w}_t) - \nabla \psi_{t-m}(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^* \rangle + \eta_t^2 \|\nabla \psi_{t-m}(\mathbf{w}_{t-m})\|_2^2, \overset{(2)}{\leq} (1 - \eta_t\lambda)\|\mathbf{w}_t - \mathbf{w}^*\|_2^2 - 2\eta_t \langle \nabla \psi_{t-m}(\mathbf{w}_t) - \nabla \bar{\psi}_{t-m}(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle + \eta_t^2 Ru\sqrt{2u^2 + 2\lambda^2R^2}m(\log m + 2) + \eta_t^2(2u^2 + 2\lambda^2R^2)$.

Above, (1) uses the fact that $-(\nabla \bar{\psi}_{t-m}(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle) \leq -(\bar{\psi}_{t-m}(\mathbf{w}_t) - \bar{\psi}_{t-m}(\mathbf{w}^*) + \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{w}^*\|_2^2)$ from the strong convexity of $\bar{\psi}(\cdot)$ and the assumption of $\mathbf{w}^*$. (2) is because $2\eta_t \langle \nabla \psi_{t-m}(\mathbf{w}_t) - \nabla \psi_{t-m}(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^* \rangle \leq 2\eta_t \|\nabla \psi_{t-m}(\mathbf{w}_t) - \nabla \psi_{t-m}(\mathbf{w}_{t-m})\|_2 \|\mathbf{w}_t - \mathbf{w}^*\|_2 \leq 4\eta_t R \|\nabla \psi_{t-m}(\mathbf{w}_t) - \nabla \psi_{t-m}(\mathbf{w}_{t-m})\|_2 \leq \eta_t Ru \|\mathbf{w}_t - \mathbf{w}_{t-m}\|_2$, where the last inequality uses that fact that $\nabla \psi_{t-m}(\cdot)$ is $\frac{1}{4}$ smooth. To proceed, for the term $\|\mathbf{w}_t - \mathbf{w}_{t-m}\|_2$, it can fur-

ther be bounded by $\Sigma_{s=0}^{m-1}\|\mathbf{w}_{t-s} - \mathbf{w}_{t-s-1}\|_2 \leq \Sigma_{s=0}^{m-1}\|\Pi_R(\mathbf{w}_{t-s-1} - \eta_{t-s-1}\nabla f_t(\mathbf{w}_{t-s-1-m})) - \mathbf{w}_{t-s-1}\|_2 \leq \sqrt{2u^2 + 2\lambda^2 R^2}\Sigma_{s=0}^{m-1}\eta_{t-s-1} = \frac{2\sqrt{2u^2+2\lambda^2R^2}}{\lambda t}(\frac{t}{t-1} + \cdots + \frac{t}{t-m}) = \frac{2\sqrt{2u^2+2\lambda^2R^2}}{\lambda t}(m + \Sigma_{s=1}^{m}\frac{s}{t-s}) \leq \frac{2\sqrt{2u^2+2\lambda^2R^2}}{\lambda t}(m + m\Sigma_{s=1}^{m}\frac{1}{t-s}) \leq \frac{2\sqrt{2u^2+2\lambda^2R^2}}{\lambda t}m(\log m + 2) = \eta_t\sqrt{2u^2 + 2\lambda^2R^2}m(\log m + 2)$. Therefore, $2\eta_t\langle\nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^*\rangle \leq \eta_t^2 Ru\sqrt{2u^2 + 2\lambda^2R^2}m(\log m + 2)$.

Now, by following the proof of Theorem 1, we can derive the result. □

Using Theorem 3, we can derive the regret of the distributed algorithm.

Theorem 4. *Let $\delta \in (0, 1/e)$ and assume $T \geq 4$ and $\lambda \leq 1$. Denote $a = \frac{(2304u^2+96uR)\log(T\log(T)/\delta)+24(u^2+\lambda^2R^2)}{\lambda^2}$ and $b = \frac{12Ru\sqrt{2u^2+2\lambda^2R^2}m(\log m+2)}{\lambda^2}$. Set $\gamma_t \leq (a+b)\frac{1}{t}$. Then, with probability at least $1-\delta$, our distributed algorithm achieves $T\max_{x\in\mathcal{D}}\mathbf{x}^\top\mathbf{w}^* - \Sigma_{t=1}^{T}\mathbf{x}_t^\top\mathbf{w}^* \leq 4\sqrt{(a+b)T\log(T)}$.*

Recall that T is the index maintained by the master, and it is the total number of rounds conducted by all the m learners. That is, each learner conducts T/m rounds, assuming T is a multiple of m. According to the theorem, if the quantity $a \gg b$, then the term b, which is introduced by the delay, is negligible. The meaning is that each learner has $\mathcal{O}(\sqrt{m})$ speedup in learning compared to the case when learning alone; if each learner independently processes T/m rounds without communication, then the total regret would be $m\tilde{\mathcal{O}}(\sqrt{T/m}) = \tilde{\mathcal{O}}(\sqrt{Tm})$, while in our case the total regret is $\tilde{\mathcal{O}}(\sqrt{T})$. The condition for $\mathcal{O}(\sqrt{m})$ speedup holds when m is not too large. On the other hand, if both quantities are at the same order, then the total regret would be $\tilde{\mathcal{O}}(\sqrt{Tm})$ regret which means no speedup is achieved, compared to the performance of learning without communication. We note that the analysis can be seen as a case that a learner cannot receive the feedback right before the end of each round, but receives it m rounds later. Back to the distributed setting, though the master equivalently experiences the delayed feedbacks, many rounds can be conducted simultaneously during an interval. This is why the parallelization can help each learner to have a better regret.

Experiment

In the experiments, we compare our algorithm (Algorithm 1) with (Abbasi-yadkori et al. 2011), the popular stochastic linear bandit algorithm. The experiments are conducted on two datasets.

The first dataset is Yahoo! Webscope dataset (R6A) [1]. It is the benchmark of measuring and comparing performance of bandit algorithms (Chu et al. 2009; Li et al. 2011). Each line in the log files represents a user interacting with one randomly chosen article from a pool of articles. It records a click ($r_t = 1$) or no-click ($r_t = 0$) for the recommended article. The articles available to present to a user in each round is the subset of the articles. That is, the decision set $\mathcal{D}_t$ may be different over time. Each record in the log file is obtained by randomly and uniformly selecting an available article for recommendation. The way of collecting the records can be used to construct an unbiased estimator of the performance for a bandit algorithm. (Li et al. 2011) suggests an algorithm being evaluated to step through the log files line by line. If the algorithm recommends the same article as the one recorded in the line, the event is added to the history and the algorithm is updated; otherwise, it just simply ignore the line. The measure of the performance is CTR score, defined as the number of clicks divided by the number of retained events (records).

Table 1: Performance of the baseline on Yahoo! R6A dataset.

feature dimensions	baseline CTR	baseline average running time (s)
36 (order=1)	5.211 %	5.95×10^{-4}
576 (order=4)	5.350 %	9.07×10^{-2}
3600 (order=10)	n/a	3.40

Table 2: Performance of Algorithm 1 on Yahoo! R6A dataset.

feature dimensions	Algorithm 1 CTR	Algorithm 1 average running time (s)
36 (order=1)	4.876 %	1.36×10^{-4}
576 (order=4)	5.533 %	7.00×10^{-3}
3600 (order=10)	5.187 %	3.62×10^{-2}

Each user and each article in the Yahoo! R6A dataset is represented by a 6 dimensional feature vector. Based on the raw feature vectors, we construct some rich feature representation for the available articles in each round. The way we form the high-dimensional features is described as follows. Denote a raw vector as $\mathbf{v} = [v[1], v[2], \ldots, v[6]]^\top$. We can generate the m_{th} order representation as $\mathbf{u} = [v[1], v[2], \ldots, v[6], v[1]^2, v[2]^2, \ldots, v[6]^2, \ldots, v[1]^m, v[2]^m \ldots, v[6]^m]^\top$. Then, the final constructed feature vectors of the articles are obtained by conducting the outer product of each m_{th} order representation of a user vector and available articles' vectors in each round. For the m_{th} order, it generates a $(m \times 6)^2$ dimensional feature vector for each article.

The second dataset is MovieLens 10M dataset [2]. This dataset consists of tuples (user's ID, movie's ID, rating score [1-5]). We assume that the ratings which are less than 4 as no-click ($r_t = 0$), the other cases are click ($r_t = 1$). We try to simulate the bandit problem as the first dataset. The way we construct a pseudo log file is as follows. First, we choose the 200 movies that get most clicks and the 200 movies that get most no-clicks. The union is the set of 324 movies. Then, a tuple in the original rating file is randomly sampled and at the same time the decision space is constructed by randomly sampling 25 items from the pool of 324 movies. If the movie

[1] https://webscope.sandbox.yahoo.com/

[2] http://grouplens.org/datasets/movielens/

indicated by the sampled tuple is in the sampled decision set, then the tuple and decision space is added to the pseudo log file. The procedure is repeated to construct about 300 thousands records in the pseudo log file. For the feature representation, we use LIBMF [3], a matrix factorization toolkit, to construct the items' features based on the ratings. The feature dimension is set to 100, 500, and 1000 in the experiment. The evaluation follows the same procedure as the Yahoo! R6A dataset.

There are parameters for the baseline and our algorithm. Denote the feature vector of an available item k in round t as $\mathbf{x}_{t,k}$. For the algorithm of (Abbasi-yadkori et al. 2011), the score of each item when making a decision is computed as $\mathbf{w}_t^\top \mathbf{x}_{t,k} + \alpha_1 \sqrt{\mathbf{x}_{t,k}^\top \mathbf{M}_t^{-1} \mathbf{x}_{t,k}}$, where $\mathbf{w}_t$ is the online least squares solution and $\mathbf{M}_t$ is the matrix that facilitates exploration (please see (Abbasi-yadkori et al. 2011) for details). We set α_1 as a tuning parameter. For our algorithm, there are two parameters η_t and γ_t. For η_t, we set $\eta_t = \frac{150}{t}$ for the MovieLens dataset, and set $\eta_t = \frac{\alpha_2}{t}$ for the Yahoo! R6A dataset, where α_2 is a tuning parameter. For γ_t, we set $\gamma_t = \alpha_3 \frac{(2304u^2+96uR)\log(T\log(T)/\delta)+24(u^2+\lambda^2R^2)}{\lambda^2 t}$, where α_3 is a tuning parameter. For λ, we simply set it to 0. In both algorithms, when making a decision, computing the score of each item is parallelizable (e.g. line 1 in Algorithm 1 in our case). That is, a number of threads can be created and each thread can compute the scores for some items at the same time. We use OpenMP/C++ to achieve that. Codes to reproduce the experiments will be available online.

Experiment results on Yahoo! Webscope dataset (R6A)

The algorithms are executed on the 05/01/2009 log file in the dataset, which consists of 4 million records. The original random policy for collecting the records achieves 3.10 % CTR. Table 1 and Table 2 show the performance of the baseline and our algorithm respectively. Since the number of retained records (i.e. number of effective rounds) by each algorithm during evaluation is different, the average running time in each round for retained records is reported

As we can see from the tables, our algorithm is significantly faster than the baseline. Moreover, the CTR scores of our algorithm are highly competitive with the baseline. For the tenth order feature representation, the baseline cannot finish the experiment in three weeks, so we could not provide the CTR (n/a). The tables also show that there are some improvements in CTR using high dimensional feature vectors. The CTR scores of both algorithms are higher when using the fourth order features, compared to the ones using the first order features. Yet, when using a much higher feature vectors (i.e. order=10), the improvement is degraded to some degree. However, as the dimension of raw features of this dataset is 6, constructing high dimensional features may be hard.

Experiment results on MovieLens dataset

Table 3 and Table 4 show the performance of the baseline and our algorithm respectively. In this dataset, a policy that randomly and uniformly choose an item in each round has 0.65 CTR. From the tables, we see that our algorithm is better than the baseline measured by both CTR and running time.

[3] https://www.csie.ntu.edu.tw/~cjlin/libmf/

Table 3: Performance of the baseline on MovieLens dataset.

feature dimensions	baseline CTR	baseline average running time (s)
100	0.8549	1.46×10^{-2}
500	0.8640	2.80×10^{-1}
1000	0.8579	2.08

Table 4: Performance of Algorithm 1 on MovieLens dataset.

feature dimensions	Algorithm 1 CTR	Algorithm 1 average running time (s)
100	0.8657	1.40×10^{-3}
500	0.8697	4.40×10^{-3}
1000	0.8673	1.44×10^{-2}

Simulation for distributed bandit

Figure 1 shows the simulation results for our distributed algorithm on Yahoo! R6A dataset, which are the CTRs with respect to number of rounds per learner under different values of parameter m. We can see that the gain increases with number of learners m till some points (m=50). Yet, even for $m = 100$, the learning rate of each learner is much faster than the one of learning without communication ("no delay" in the figure). This demonstrates that our algorithm works well in the distributed setting.

Conclusion

In this paper, we propose an efficient algorithm whose update is fast and regret upper bound does not depend explicitly on the dimension of the context space. Our algorithm admits using high dimensional context vectors, which offers flexibility for feature engineering. The significance is reflected on the benchmark dataset. Furthermore, we develop a distributed algorithm and analyze its regret. We believe that it is a big step towards developing distributed bandit algorithms that work well in applications with theoretical guarantees. Future works include implementing our distributed algorithm in a real recommendation system as well as considering different communication protocols.

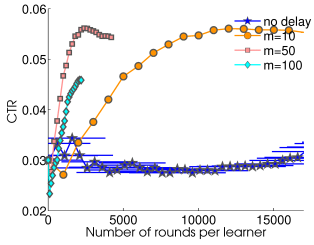

Figure 1: Simulation for distributed bandit.

References

[Abbasi-yadkori et al. 2011] Abbasi-yadkori, Y.; Pál, D.; Garivier, A.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. *NIPS*.

[Agarwal and Duchi 2011] Agarwal, A., and Duchi, J. 2011. Distributed delayed stochastic optimization. *NIPS*.

[Auer 2002] Auer, P. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3:397–422.

[Chu et al. 2009] Chu, W.; Park, S.; Beaupre, T.; Motgi, N.; Phadke, A.; and Chakraborty, S. Zachariah, J. 2009. A case study of behavior-driven conjoint analysis on yahoo!: front page today module. *KDD*.

[Dani, Hayes, and Kakade 2008] Dani, D.; Hayes, T.; and Kakade, S. 2008. Stochastic linear optimization under bandit feedback. *COLT*.

[Langford, Smola, and Zinkevich 2009] Langford, J.; Smola, A.; and Zinkevich, M. 2009. Slow learners are fast. *NIPS*.

[Li et al. 2010] Li, L.; Chu, W.; Langford, J.; and Schapire, R. 2010. A contextual-bandit approach to personalized news article recommendation. *WWW*.

[Li et al. 2011] Li, L.; Chu, W.; Langford, J.; and Wang, X. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. *WSDM*.

[Rakhlin, Shamir, and Sridharan 2012] Rakhlin, A.; Shamir, O.; and Sridharan, K. 2012. Making gradient descent optimal for strongly convex stochastic optimization. *ICML*.

[Zhang et al. 2016] Zhang, L.; Yang, T.; Jin, R.; and Zhou, Z. 2016. Online stochastic linear optimization under one-bit feedback. *ICML*.